

# SLAM을 위한 경량화 특징점 추출 모델 설계 및 실험

임은지<sup>1</sup>, 이승우<sup>1</sup>, 강버들<sup>1</sup>, 정해찬<sup>1</sup>, 김재수<sup>1</sup>, 조재진<sup>2</sup>, 이철희<sup>3</sup>

<sup>1</sup>경북대학교 컴퓨터학부 학부생, 석사과정, 교수

<sup>2</sup>경북대학교 물리교육학과 학부생

<sup>3</sup>(주)제이씨레이다 대표

eunzi@knu.ac.kr, tmdrhdqn@knu.ac.kr, beodeul0@knu.ac.kr,

sunishae44@knu.ac.kr, kjs@knu.ac.kr, jycho1021@knu.ac.kr, cherlhee@jcradar.com

## Lightweight Feature Extractor for SLAM Design and Evaluation

Eun-ji Lim<sup>1</sup>, Seung-woo Lee<sup>1</sup>, Beo-deul Kang<sup>1</sup>,  
Hae-chan Jeong<sup>1</sup>, Jae-soo Kim<sup>1</sup>, Jae-jin Cho<sup>2</sup>, Cheol-hee Lee<sup>3</sup>

<sup>1</sup>Dept. of Computer Science, Kyung-Pook University

<sup>2</sup>Dept. of Physics Education, Kyung-Pook University

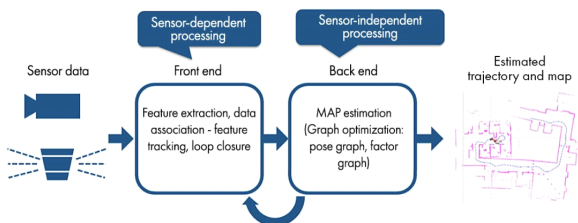
<sup>3</sup>CEO, JC Radar Co., Ltd

### 요 약

물류 로봇의 확산에 따라 SLAM(Simultaneous Localization and Mapping)의 중요성이 더욱 부각되고 있다. SLAM 프론트엔드 단계의 특징점 추출 및 디스크립터 생성은 시스템 성능에 직접적인 영향을 미친다. 기존의 ORB는 연산 효율은 높으나 정확도에 한계가 있으며, SuperPoint는 정확도는 우수하나 연산 자원이 많이 요구된다. 본 논문에서는 SuperPoint를 경량화하여 저사양 임베디드 환경에 적합한 구조로 설계하고, 실험을 통해 성능을 검증하였다. 그 결과, 제안된 모델은 정확도와 효율 측면에서 균형을 이루며 산업적 활용 가능성을 보여주었다.

### 1. 서론

최근 자율주행 로봇과 물류 자동화 수요가 증가하면서, 공간 인식 및 경로 계획의 핵심 기술인 SLAM(Simultaneous Localization and Mapping)에 대한 관심도 높아지고 있다. SLAM은 센서 데이터를 처리하는 프론트엔드와, 위치 추정 및 지도 생성을 수행하는 백엔드로 구성되며(그림 1), 이 중에서도 프론트엔드 단계의 특징점 추출과 디스크립터 생성은 시스템의 정확도와 안정성에 직접적인 영향을 미친다. 특히, 연산 자원이 제한된 임베디드 환경에서도 안정적으로 동작할 수 있는 특징점 추출 기법의 중요성이 점차 부각되고 있다.



(그림 1) SLAM의 처리 흐름[1]

전통적으로 ORB(Oriented FAST and Rotated BRIEF)와 같은 특징점 기반 알고리즘은 계산량이 적고 실시간 처리가 가능하다는 장점으로 널리 사용되어 왔다. 하지만 조명 변화나 회전에 대한 강인성이 부족하다는 한계가 있으며, 복잡한 환경에서는 정확도가 크게 저하될 수 있다. 이를 보완하기 위해 제안된 SuperPoint 알고리즘은 딥러닝 기반의 특징점 검출 및 기술 방법으로, 높은 정밀도와 반복성(repeatability)을 제공하지만, 연산량과 메모리 요구가 커 저사양 환경에서의 활용에 제약이 따른다.

본 연구는 이러한 한계를 해결하고자, SuperPoint의 구조를 경량화하여 저사양 임베디드 시스템에서도 실시간 처리가 가능한 Lightweight SuperPoint 모델을 설계하고, Jetson Nano 환경에서 그 실효성을 실험적으로 검증하였다.

### 2. 관련 연구

#### 2.1 ORB 알고리즘

ORB는 FAST 기반의 특징점 검출기와 BRIEF 디스크립터를 결합한 알고리즘으로, 계산량이 적어

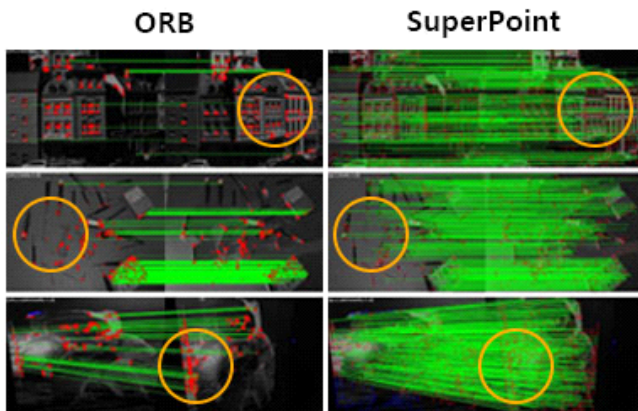
저사양 환경에서도 빠르고 효율적인 특징점 추출이 가능하다[2]. 특히 연산 자원이 제한된 임베디드 시스템에서 널리 활용되고 있다.

## 2.2 SuperPoint 알고리즘

SuperPoint는 딥러닝 기반의 특징점 검출 및 디스크립터 생성 모델로, 학습을 통해 높은 정확도를 보여준다. 하드웨어 성능의 발전과 함께 실시간 처리도 가능해지면서, 최근의 SLAM 및 비전 기반 응용 분야에서 대표적인 딥러닝 특징점 알고리즘으로 자리 잡았다.

## 2.3 기존 연구의 한계

기존 SLAM 시스템에서는 ORB와 같은 전통적인 알고리즘이 널리 사용되어 왔지만, 조명 변화나 회전에 대한 강인성이 낮고 정확도가 부족하다는 한계가 있다. 이를 보완하기 위해 등장한 SuperPoint는 데이터 기반 학습을 통해 복잡한 환경에서도 높은 정확도와 견고한 성능을 보여준다. DeTone et al. [3]의 연구에 따르면, SuperPoint는 ORB에 비해 특징점 검출에서 우수한 결과를 나타낸다.



(그림 2) ORB와 SuperPoint의 특징점 검출 결과 및 매칭 성능 비교  
기존 연구 [3]의 이미지를 참고하여 편집·재구성함.

이러한 성능의 차이는 실제 이미지에서의 매칭 결과를 통해 더욱 명확하게 확인할 수 있다. (그림 2)는 동일한 이미지를 ORB와 SuperPoint 알고리즘으로 처리한 뒤, 추출된 특징점을 기반으로 좌·우안 이미지를 매칭한 결과를 보여준다. 주황색 원은 동일한 물체를 기준으로 시각적 매칭 성능을 비교하기 위해 설정된 영역으로, SuperPoint는 해당 영역에서 안정적인 매칭을 보여주는 반면, ORB는 특징점을

비교적 잘 검출하였음에도 불구하고 동일 물체의 좌·우 특징점을 정확히 연결하지 못해 매칭 성능이 저하된 모습을 확인할 수 있다. 특히, 복잡한 구조물이나 질감이 적은 영역에서는 이러한 한계가 더욱 두드러진다.

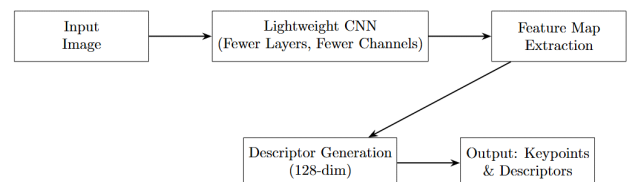
SuperPoint는 더 많은 특징점을 안정적으로 검출하고, 동일 물체에 대한 좌·우 매칭 정확도 또한 높아 우수한 성능을 시각적으로 입증하였다. 이러한 결과는 SuperPoint가 다양한 환경에서도 더욱 견고한 특징점 추출 및 매칭 성능을 제공할 수 있음을 보여준다.

SuperPoint는 높은 정확도를 제공하지만, 연산량과 메모리 사용이 많아 저사양 임베디드 환경에서 실시간 활용에 제약이 따른다. 동일한 환경에서 실행한 결과, ORB는 약 200MB 이하의 메모리를 사용하는 반면, SuperPoint는 약 500MB 수준의 메모리를 요구하여 상대적으로 높은 연산 자원을 소모하는 것으로 나타났다. 이러한 이유로 SuperPoint는 연산 자원이 제한된 임베디드 시스템에서의 적용에 어려움이 있다.

## 3. 제안 기법

본 연구에서는 SuperPoint의 구조를 간소화하여 연산량과 메모리 사용을 줄인 Lightweight SuperPoint 모델을 설계하였다. 해당 모델은 저사양 임베디드 환경에서도 실시간 특징점 추출이 가능하도록 설계되었으며, 다양한 실험을 통해 성능을 평가하였다.

Lightweight SuperPoint의 처리 과정은 (그림 3)의 흐름도를 통해 확인할 수 있다. 입력 이미지는 경량화된 CNN 구조를 통해 처리되며, 특징점과 함께 128차원으로 축소된 디스크립터가 동시에 생성된다. 이러한 경량화 설계는 전체 연산량을 줄이고, 저사양 환경에서도 실시간 처리를 가능하게 한다.



(그림 3) Lightweight SuperPoint 모델의 처리 구조 흐름도

### 3.1 실험 환경 설정

경량 모델의 성능을 저사양 환경에서 평가하기 위해 NVIDIA Jetson Nano 4GB 보드의 사양을 기준으로 실험 환경을 구성하였다. Jetson Nano는 가격 대비 성능이 우수하고, SLAM 기반 로봇 시스템에 많이 활용되는 대표적인 임베디드 장비이다.

실험은 Jetson Nano와 유사한 사양을 갖춘 WSL(Windows Subsystem for Linux) 기반 Ubuntu 환경에서 수행되었으며, GPU 없이 CPU만을 사용하여 모델을 실행하였다. 이러한 구성은 실제 저사양 장비에서의 사용 가능성을 평가하는 동시에, 실험 접근성을 높이기 위한 목적을 가진다.

### 3.2 Lightweight SuperPoint 설계 및 학습

기존 SuperPoint 모델은 총 13개의 convolutional layer와 256차원의 디스크립터로 구성되어 있으며, 특징점 검출과 디스크립터 생성을 통합한 구조로 높은 정확도를 제공한다 [3].

그러나 이러한 복잡한 구조는 연산 자원이 제한된 환경에서 실시간 처리를 어렵게 만드는 주요 원인으로 작용한다.

본 연구에서는 다음과 같은 방식으로 SuperPoint의 구조를 경량화하였다:

- 채널 수 축소: 각 convolution 레이어의 채널 수를 절반 이하로 줄여 연산량을 효과적으로 감소시켰다.
- 레이어 수 축소: 중복 연산 구조를 제거하고 병목을 유발하는 블록을 단순화하여 모델의 깊이를 축소하였다.
- 디스크립터 차원 축소: 기존 256차원의 디스크립터를 128차원으로 줄여 메모리 사용량을 절감하고 매칭 속도를 향상시켰다.

<표 1> SuperPoint와 Lightweight SuperPoint 구조 비교

	SuperPoint	Lightweight SuperPoint
채널 수	64 → 128 → 256	32 → 64 → 128
레이어 수	13	9
디스크립터 차원	256차원	128차원

<표 1>은 제안한 모델이 기존 SuperPoint에 비해 채널 수, 레이어 수, 디스크립터 차원 면에서 구조적으로 더욱 간결해졌음을 보여준다. 이로 인해 연산량과 메모리 사용이 줄었고, 특히, 디스크립터 차원 축소는 매칭 속도 향상에 크게 기여하였다.

이러한 구조적 단순화는 모델의 처리 흐름 전반에도 영향을 주었다.

Lightweight SuperPoint는 입력 이미지로부터 히트맵(heatmap)을 생성해 각 위치의 특징점 존재 확률을 예측하고, NMS(Non-Maximum Suppression)를 통해 신뢰도 높은 특징점만을 선택한다. 이후 각 위치에서 128차원의 디스크립터를 병렬 경로를 통해 추출하며, 전체 모델은 end-to-end 방식으로 학습된다.

실제 이미지 기반의 일반화 성능을 높이기 위해 COCO 2017 train 데이터셋을 활용하여 학습을 수행하였다. 총 15 epoch 동안 학습을 진행한 결과, 평균 손실값(loss)은 약 0.0078 수준에서 수렴하였다.

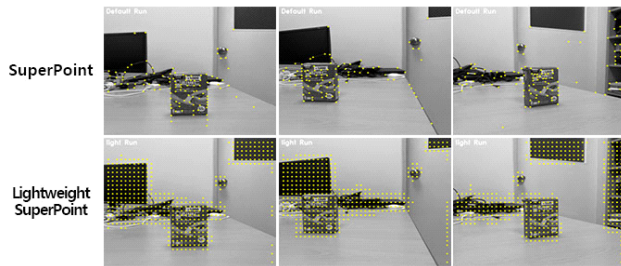
추가로, 선·도형 등 기하학적 구조에 특화된 특징점 인식을 강화하기 위해 직접 생성한 synthetic shapes 데이터셋을 사용하였다. 해당 학습은 10 epoch 동안 진행되었으며, 평균 손실값은 약 0.0079를 기록하였다.

이처럼 실제 이미지와 synthetic 데이터를 함께 활용함으로써 다양한 환경에서도 안정적으로 동작하는 특징점 검출 모델을 구현할 수 있었으며, 범용성과 정밀도 또한 향상되었다.

### 3.3 평가

제안한 Lightweight SuperPoint는 기존 SuperPoint에 비해 모델 파라미터 수와 연산량을 크게 줄이면서도, 주요 물체에 대한 특징점 검출 성능은 유사한 수준을 유지하였다. 성능 검증은 학습에 사용되지 않은 COCO val2017 데이터셋 중 사람 객체가 포함되지 않은 30장의 이미지를 선별하여 진행되었으며, 평균 652.30개의 특징점이 검출되었으며, 이미지 한 장당 평균 추론 시간은 약 1.1514초로 측정되었다.

다양한 환경과 객체를 포함한 테스트 이미지에서도 안정적인 검출 성능을 유지하였으며, 특히 윤곽선, 외곽 경계, 텍스처가 뚜렷한 영역에 특징점이 균일하게 분포하는 경향을 보였다. 추가적으로, 하나의 물체를 다양한 각도에서 촬영한 이미지에 대해 두 모델을 실행한 결과, 기존 SuperPoint는 물체의 일부 경계나 꼭짓점에 집중된 반면, Lightweight SuperPoint는 물체 전체 영역에 특징점이 고르게 분포하는 형태로 나타났으며, 이는 (그림 5)에서 확인된다. 제안한 모델이 2D 기반의 로컬라이징 환경에서도 유의미한 성능을 갖춤을 나타낸다.



(그림 4) SuperPoint와 Lightweight SuperPoint  
특징점 추출 결과

<표 2> SuperPoint와 Lightweight SuperPoint  
성능 비교

	SuperPoint(평균)	Lightweight SuperPoint(평균)
특징점 수	105.7	411.7
추론 시간	95.51ms	64.64ms
모델 크기	5.0MB	1.3MB
메모리 최대값	524.63MB	477.48MB

모델 크기는 약 1/4로 줄었고, 추론 시간은 기존 대비 약 1.5배 가까이 단축되면서 경량화의 효과가 분명히 나타났다. 이러한 결과는 연산 자원이 제한된 환경에서도 딥러닝 기반 특징점 추출 모델이 실시간으로 활용 가능함을 입증하였다. 본 연구에서 제안한 경량화 접근법은 SLAM 시스템의 실용적 구현에 기여할 수 있음을 시사하며, 향후 다양한 임베디드 환경에서의 확장 가능성도 기대할 수 있다.

또한, 제안한 모델은 기존보다 더 많은 수의 특징점을 추출함으로써 물체 전반에 대한 정보 손실을 줄이고, 매칭 실패 가능성을 낮출 수 있는 장점이 있다. 이는 반복성 향상에도 긍정적인 영향을 미칠 수 있으며, 복잡한 환경에서도 안정적인 맵 구성을 지원한다는 측면에서 유의미한 결과라 할 수 있다. 다만, 특징점 수의 증가가 후속 연산의 부담으로 작용할 수 있으므로, 전체 시스템 차원의 균형 있는 설계가 요구된다.

#### 4. 결론

본 연구에서는 SuperPoint의 정확도를 유지하면서도 경량화를 통해 임베디드 환경에 적합한 Lightweight SuperPoint 모델을 설계하였다. 실험 결과, 제안된 모델은 기존 SuperPoint에 비해 약 1.5배 빠른 추론 속도와 약 1/4 수준의 모델 크기를 달성하면서도, 주요 물체에 대한 특징점 검출 성능을 효과적으로 유지하였다. 이러한 성능은 GPU 없이도

연산 자원이 제한된 환경에서 안정적인 실시간 처리를 가능하게 하며, 임베디드 기반의 SLAM 시스템 구현에 효과적임을 시사한다. 특히, 저전력 모바일 로봇이나 무인 운송 시스템 등 다양한 응용 환경에서 실질적인 이점을 제공할 수 있다.

또한, 실제 환경뿐만 아니라 도형 기반 synthetic 데이터에서도 안정적인 학습 성능을 보였으며, 학습 데이터의 다양성과 경량화 구조의 설계가 모델의 일반화 성능 향상에 기여했음을 확인할 수 있었다.

향후에는 조도가 높은 환경이나 밝은 배경 등 극단적인 조건에서도 성능 저하 없이 동작할 수 있도록, 조명 변화에 강인한 학습 전략이나 후처리 기법을 도입할 예정이다. 이러한 개선은 극한 환경에서도 모델의 활용 가능성을 더욱 확대시킬 수 있을 것으로 기대된다.

#### 참고문헌

- [1] MathWorks, "What Is SLAM?", MathWorks 공식 홈페이지, 2024. [Online]. Available: <https://kr.mathworks.com/discovery/slam.html>
- [2] E. Rublee et al., "ORB: An efficient alternative to SIFT or SURF," Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011, pp. 2564 - 2571.
- [3] D. DeTone et al., "SuperPoint: Self-Supervised Interest Point Detection and Description," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, USA, 2018, pp. 224 - 236.